

Claims

5 1. A method for testing a software product in a distributed testing system, the distributed testing system including a main server component system and at least two client component systems, comprising:

generating at least one data object to be locally stored in a first location of a first memory of the first client component system;

10 registering the first location with the main server component system; and

transferring the data object from the first memory of the first client component system to a second memory of the second client component system; the transferring being in response to the second client component system requesting the data object from the first client component system through the main server component system.

15 2. A method for testing a software product in a distributed testing system as recited in claim 1, wherein the registering the first location further comprising,

storing the first location in a shared object table of the main server component system.

20 3. A method for testing a software product in a distributed testing system as recited in claim 1, further comprising,

registering the first location with the first client component system.

4. A method for testing a software product in a distributed testing system as recited in claim 3, wherein the registering the first location further comprising,

5 storing the first location in a client table of the first client component system.

5. A method for testing a software product in a distributed testing system as recited in claim 1, wherein the registering the first location with the main server component system is defined by one of a put function and a putb function.

10 6. A method for testing a software product in a distributed testing system as recited in claim 1, wherein the data object is defined by one of a resulting test data generated after executing a portion of a test at the first client component system, an identification key, a dynamically generated Java class, a configuration file, a property file, and an initial test
15 data transmitted by the main server component system to each of the first and second client component systems to initialize the test.

7. A method for testing a software product in a distributed testing system as recited in claim 6, wherein the identification key is used in a secured protocol to establish a
20 secured communication between the main server component system and each of the first and second client component systems.

8. A method for testing a software product in a distributed testing system as recited in claim 1, wherein the each of the first and second memory is defined by one of a random-access memory (RAM), a dynamic RAM (DRAM), and a static RAM (SRAM).

5

9. A method for testing a software product in a distributed testing system as recited in claim 1, wherein the first location is a memory address of the first memory.

10. A method for testing a software product in a distributed testing system as
10 recited in claim 1, wherein the second client component system requesting the data object from the first client component system is defined by one of a get function and a getb function.

11. A method for testing a software product in a distributed testing system as
15 recited in claim 1, further comprising,

executing a distributed test harness on the main server component system; and

executing a client harness on each of the first and second client component systems.

12. A method for testing a software product in a distributed testing system, the
20 distributed testing system including a main server component system and at least two client component systems, comprising:

executing a portion of a test at a first client component system, the executing
generating at least one data object to be locally stored in a first location of a first memory of
the first client component system;

5 registering the first location with the main server component system; and

 transferring the data object from the first memory of the first client component
system to a second memory of the second client component system; the transferring being in
response to the second client component system requesting the data object from the first
client component system through the main server component system.

10

13. A method for testing a software product in a distributed testing system as
recited in claim 12, wherein the registering the first location further comprising,

 storing the first location in a shared object table of the main server component
system.

15

14. A method for testing a software product in a distributed testing system as
recited in claim 12, further comprising,

 registering the first location with the first client component system.

20

15. A method for testing a software product in a distributed testing system as
recited in claim 14, wherein the registering the first location further comprising,

 storing the first location in a client table of the first client component system.

16. A method for testing a software product in a distributed testing system as recited in claim 12, wherein the registering the first location with the main server component system is defined by one of a put function and a putb function.

17. A method for testing a software product in a distributed testing system as recited in claim 12, wherein the each of the first and second memory is defined by one of a random-access memory (RAM), a dynamic RAM (DRAM), and a static RAM (SRAM).

18. A method for testing a software product in a distributed testing system as recited in claim 12, wherein the first location is a memory address of the first memory.

19. A method for testing a software product in a distributed testing system as recited in claim 12, wherein the second client component system requesting the data object from the first client component system is defined by one of a get function and a getb function.

20. A method for testing a software product in a distributed testing system as recited in claim 12, further comprising,
executing a distributed test harness on the main server component system; and
executing a client harness on each of the first and second client component systems.

21. A system for testing a software product in a distributed testing system,
comprising:

5 a main server component system;

a first client component system, the first client component system being in
communication with the main server component system;

a data object located in a memory of the first client component system;

a second client component system, the second client component system being in

10 communication with the main server component system; and

wherein the main server component system facilitates communication between each
of the first and second client component systems, and wherein a location of the data object is
registered with the main server component system for use in transferring the data object from
the first client component system to the second client component system.

15 22. A system for testing a software product in a distributed testing system as
recited in claim 21, further comprising,

a distributed test harness executed on the main server component system;

a first client harness executed on the first client component system; and

20 a second client harness executed on the second client component system.

23. A system for testing a software product in a distributed testing system as recited in claim 21, wherein the main server component system includes a shared object table for storing the location of the data object.

5

24. A system for testing a software product in a distributed testing system as recited in claim 21, wherein the first client component system includes a client table for storing the location of the data object.

10

25. A system for testing a software product in a distributed testing system as recited in claim 21, further comprising a data transfer monitor facility coupled to the main server component system.

15

26. A system for testing a software product in a distributed testing system as recited in claim 21, wherein the data object is defined by one of a resulting test data generated after executing a portion of a test at the first client component system, an identification key, a dynamically generated Java class, a configuration file, a property file, and an initial test data transmitted by the main server component system to each of the first
20 and second client component systems to initialize the test.

27. A system for testing a software product in a distributed testing system as recited in claim 26, wherein the identification key is used in a secured protocol to establish a

secured communication between the main server component system and each of the first and second client component systems.

5 28. A system for testing a software product in a distributed testing system as recited in claim 21, wherein the memory is defined by one of a random-access memory (RAM), a dynamic RAM (DRAM), and a static RAM (SRAM).

 29. A system for testing a software product in a distributed testing system as
10 recited in claim 21, wherein the location is a memory address of the memory.